precisely true and fulfills the standard of embedded systems definition. Table 2.1 outlines several SBCs from various manufacturers with CPU architecture, form factor and its features. Only a few examples are taken from original sources (Baxter, 2001), and with different table view.

Manufacturer/ SBC (model)	CPU Architecture/ Form Factor	Features
Motorola/ MVME5100	PowerPC/ VMEbus	750/7400 Altivec, dual-PCI mezzanine card sites, up to 1GB ECC SDRAM, dual Ethernet ports, two serial ports, up to 16MB Flash
Zynx/ ZX4500	PowerPC/ CompactPCI	24 10/100 Ethernet ports, two Gigabit Ethernet ports, PMC/PPMC slot for additional I/O and an expansion processor, fully hot-swap compliant
Ampro/ Little Board/P5x	x86/ EBX	PC/104-plus expandable PCI/ISA bus, P5x supports up to 256MB DRAM with bootable Compact Flash socket and 10/100Base-T Ethernet, USB, IrDA, KB, floppy, IDE, serial and parallel I/O, also supports C&T 69000-series PCI LCD/CRT controller with PanelLink, LVDS and NTSC options
WinSystems	x86/ PC/104	133MHz 586DX with up to 72MB Flash disk, CRT/LCD display video controller, Ethernet, IDE and floppy disk controllers, serial, parallel and keyboard
Bright Star/ mediaEngine	StrongARM/ 5.2"x5.3"	8-64MB SDRAM at 100MHz, 1-20MB Flash, Type II Compact Flash socket, Type I/II/II PCMCIA socket, 10Base-T Ethernet, three serial ports, V.90 modem, LCD panel controller, USB slave interface
Intel/ Assabet	StrongARM/ 2.5"x5"	64-256MB of TSOP SDRAM, 64-128MB onboard socketed Flash, integrated LCD support, Bluetooth, GSM digital radio, audio in and out, built-in TV encoder supporting S-video, NTSC, PAL and RGB formats, IrDA port, soft-modem support

Table 1. Embedded Linux SBCs (Baxter, 2001)

Generally the SBC is a complete computer built on a single Printed Circuit Board (PCB). It has all important elements similar to the standard computer including processor, memory and Input Output (I/O). Certain peripheral are also available within SBC including serial port, parallel port and USB port. The Ethernet port, wireless network socket, audio line in and VGA port may customize as well that are sometimes custom-built to perform specific tasks. Otherwise it does not come with default display unit and input hardware. The most

important feature of the SBC is it can run modular OS. The Z80-based "Big Board" (1980) was probably the first such SBC that was capable of running a commercial disk operating system (LinuxDevices, n.d.).
Most SBC boards use commercial off-the-shelf (COTS) processor. This helps reducing development time and dependencies on technical staff to develop dedicated processor board from scratch. The SBC processor board is suitable for use in critical and complex applications to develop a systems model or handle an analysis before running the real system such as in a flight simulator (Peters, 2007). SBCs are often integrated into dedicated equipment which is used, for example, in industrial or medical monitoring applications (James, 2000). The use of embedded systems is reasonably low cost and small physical size promising the most effective solution. It is not only suitable for portable system but also significantly improving the capabilities of the instrument (Perera, 2001). Zabolotny et al. (2003) has replaced the VME (Versa Module Eurocard bus) controller with embedded PC for TESLA cavity controller and simulator DAS. The replacement was made to enhance functionality in terms of bits and register manipulation, data processing operation and to increase efficiency of data acquisition and control and enhancing data transfer.

4. System overview

Hardware design gives an overview of the physical interaction among the devices of the system. Hardware components of the DAS are shown in Fig. 1 below. SBC acts as an acquisition hardware that acquires data from sensors. A signal conditioning circuit is used for high output impedance sensor, to match the built-in ADC on the SBC board. The developed DAS based on SBC is named Portable Embedded Sensing System (PESS). PESS is developed with an integration of SBC, matrix keypad, LCD panel and sensors. The matrix keypad functions as an input device and information data is displayed using LCD panel. Fig. 2 outlines the PESS system architecture which consists of hardware and software.

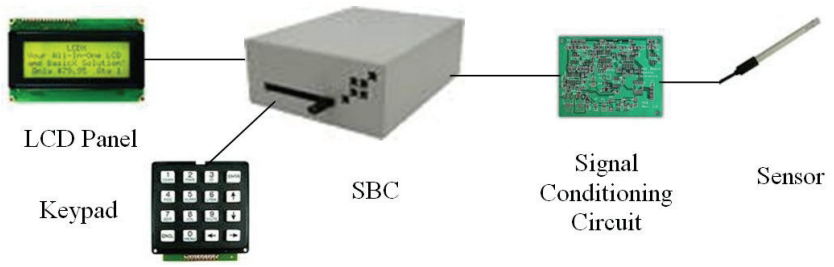


Fig. 1. PESS hardware design

The PESS system has several limitations in terms of storage capacity and data view space. Compact Flash (CF) is used as storage devices which functions as a hard disk for the SBC. The data that can be stored on the CF is up to 4GB. Due to the limitation of CF storage device, PESS is not suitable for applications that require large storage capacity.

4.1 Embedded acquisition hardware: TS-5500 SBC

Technologic System offers semi-custom and off-the-shelf Single Board Computers (SBC). The product from Technologic Systems available in two different architectures which are ARM and X86.

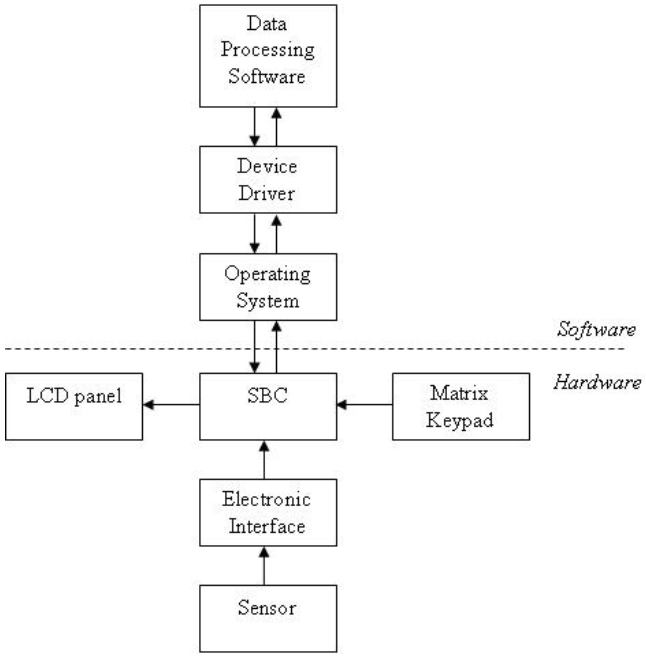


Fig. 2. PESS system's architecture

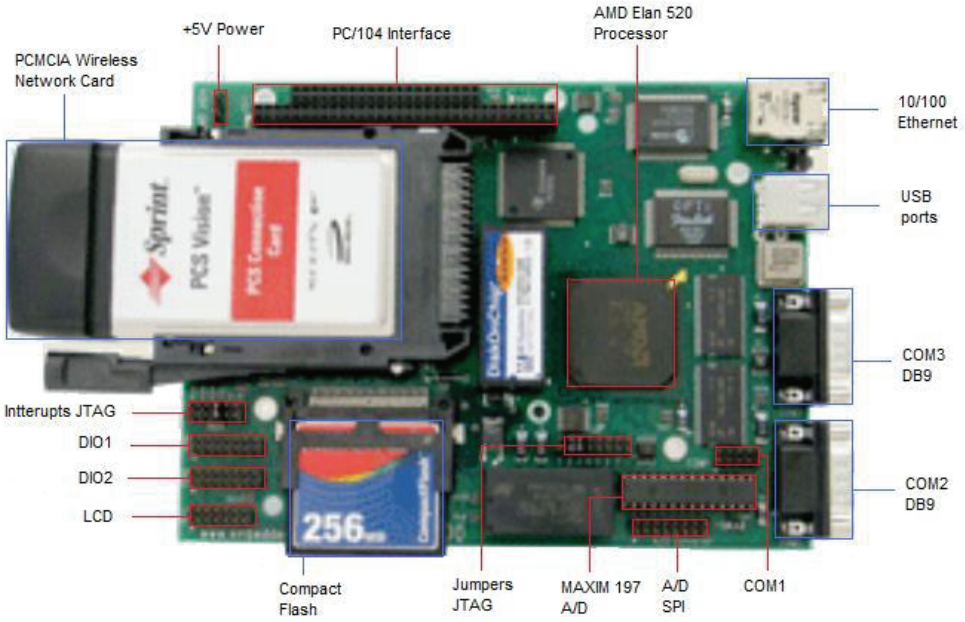


Fig. 3. TS-5500 Single Board Computer

For ARM SBCs, they can be identified with TS-7000 number series. There are four series for ARM SBCs which are TS-7200 series, TS-7300 series, TS-7400 series and TS-7800 series. The X86 SBCs is available in two series which are TS-3000 and TS-5000. The X86 SBCs have slower CPU compared to ARM SBCs. The TS-3000 series run Intel 386 CPU with 33 MHz and has small memory which is 8 MB. The TS-5000 series run 133 MHz AMD Elan 520 CPU and has 32 MB of memory. The TS-5000 series is manufactured with wireless network interface. Fig. 3 show the TS-5500 SBC main board.
TS-5500 SBC from Technologic Systems has been used by many developers in various fields including robotic, web server application and data acquisition and control system. In 2003, Hoopes, David, Norman and Helps presented the development of autonomous mobile robot based on TS-5500 SBC. The other example of robotic design and development based on TS-5500 SBC was built by Al-Beik, Meryash and Orsan.

4.2 Sensor interfacing

Two types of analog sensors are used which are temperature sensor and ion selective electrode. LM35DZ temperature sensor from National Semiconductor is a simple analog sensor used in this research where it's measurement is not using a signal conditioning circuit. Copper (Cu^{2+}) ion selective electrode from Sensor Systems are used with a reference electrode for high impedance output sensor type. Fig. 4(a) and 4(b) show the Copper ion selective electrode and reference electrode respectively.

(a)

(b)

Fig. 4. a) Copper ion-selective electrode b) Ion-selective reference electrode

The most frequently processes performed in signal conditioning are amplification, buffering, signal conversion, linearization and filtering (Ismail, 1998). ADC normally can read analog inputs that have low output impedance. If the input impedance of the sensor is high, the ADC reading is unstable and not reliable. Typically the glasses electrodes such as pH probes or gas concentration probes are of this type (Microlink, n.d.). Therefore a signal conditioning circuit has to be integrated with a high output impedance sensor (*Application notes 270*, 2000). This can be done by attaching to a voltage follower as a buffer element to match the impedance. In this research, the signal conditioning circuit built has two stages circuit. The first stage functions as a buffer unit which will decrease the input impedance from analog input. The second stage is a filter that removes the noise signal. The OPA2111 (*OPA2111*, 1993) operational amplifier is used within the signal conditioning circuit. The OPA2111 has high internal resistance of 10^{13} Ω for differential mode and 10^{14} Ω for common-mode. The signal conditioning circuit used is shown in Fig. 5.

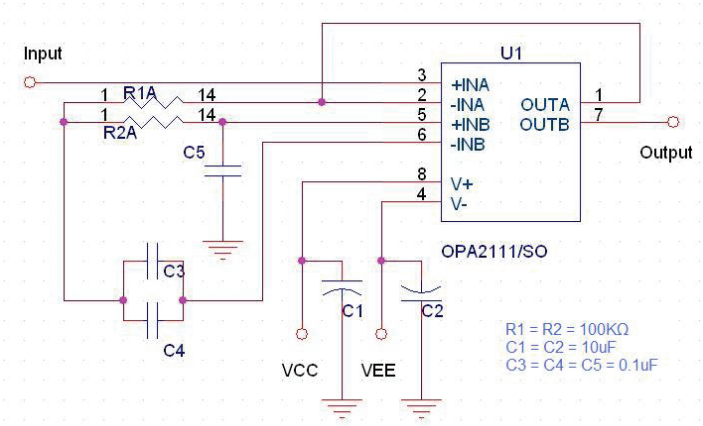
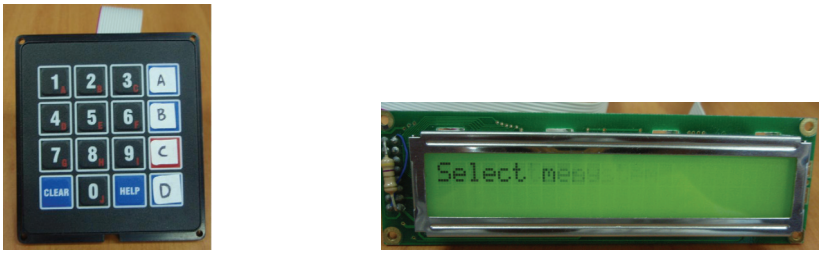


Fig. 5. Signal conditioning circuit

4.3 Input/Output of PESS system

The 4x4 16 button matrix keypad is used as input device for the system developed. The keypad is manufactured by ACT Components, Inc with physical size 4.7″W x 1.7″H x 0.4″T. A nine (9) pin input is used to connect between matrix keypad with device or processor board using serial cable. The 24x2 alphanumeric LCD panel is use as display for this system. The LCD is manufactured by Lumex Inc with physical size of 118mm x 36mm x 12.7mm. It connected to processor board using 9 inputs serial cable.



(a) (b)

Fig. 6. a) 4x4 matrix keypad b) 24x2 alphanumeric LCD panel

4.4 Embedded OS: TSLinux

Technologic Systems provides two free OSes which are developed by their research team: Linux and DOS. These OSes are developed to be used with their product only. However, many other OSes can also be used with TS products such as uC/OS-II, eRTOS, microCommander modular Human-Machine Interface (HMI), MicroDigital SMX modular and QNX Embedded Real Time OS. TSLinux is chose to run on SBC in this research. TSLinux is a PC compatible embedded Linux distribution built from open source. There is a tailored Linux kernel for each TS SBC, along with completed driver support for the hardware. The kernel source is also provided to end users to enable custom changes and development.

Several TSLinux features as follows:

- Glibc version 2.2.5
- Kernel version 2.4.18 and 2.4.23
- Apache web server with PHP
- Telnet server and client
- FTP server and client
- BASH, ASH, minicom, vi, busybox, tinylogin

5. Software development

Two software modules developed in the PESS system which are the Analog Input Preprocessing and Data Presentation. The Analog Input Preprocessing module involves data acquiring from sensor, converting analog input to digital output and calculating converted output to human readable value. A C code named *sensor* to cope all those processes is developed. Data Presentation module in PESS system is handled by a program named *Interactive System*. An *Interactive System* provides current sensor' readings and the information of the system such as disk (CF) usage and memory capacity status. Fig. 7 show the interaction between both software modules which running concurrently. *Sensor* program processing the analog inputs and store converted data into shared memory, meanwhile those current data available on shared memory can be accessed via *Interactive System* program.

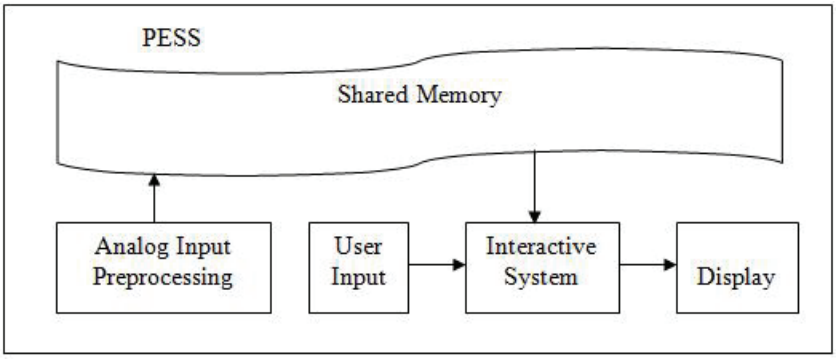


Fig. 7. Software architecture of PESS system

5.1 Analog input preprocessing

Signals from analog sensors must be converted to digital signals before electronic device can read them. The conversion from analog input to digital output is done using the ADC. The digital outputs which are in binary format is then calculated into human readable value in decimal value and presented in Volt parameter. The TS-5500 supports an eight-channel, 12-bit ADC capable of 60000 samples per second. Each channel is independently software programmable for a variety of analog input ranges: -10V to +10V, -5V to +5V, 0V to +10V and 0V to +5V. The ADC control register, the Hex 196 setting is outlined by Fig. 8 below. The IO address is read from right to left starting with 0. The settings are based on a bipolar mode with 5V output range for all channels.

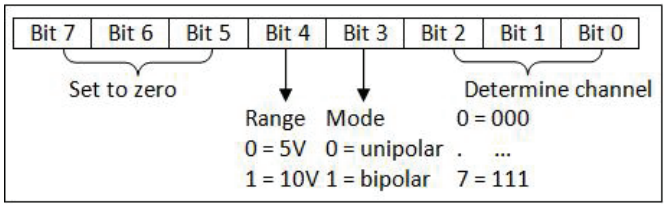


Fig. 8. ADC control register

The processes of Analog Input Preprocessing can be divided into four stages: initialization, bit checking, reading and storing. At the initialization stage, the permission to access ADC IO register must be set. Three registers are involved in accessing the ADC I/O address which are, Hex 195, Hex 196 and Hex 197. The digital output of an analog input is available after the ADC has completely converted the input within 11µs. The End of Conversion (EOC) status can be checked at bit 0 of register Hex 195. The conversion is completed if the bit 0 of Hex 195 indicates '0'. The digital output of the converted analog input is available at Hex 196 and Hex 197. 8 bits of them is available at Hex 196 which called as the lower 8 bits or LSB. The other 4 bits is available at Hex 197 which called as the upper 4 bits or MSB.

Analog Input Preprocessing algorithm

Step 1 : Initialize the IO permission of ADC
Step 2 : Create and attach shared memory file descriptor
Step 3 : Set up ADC control registers
Step 4 : Check End Of Conversion (EOC) signal
4.1 If EOC signal HIGH (1)
Go to Step 4 until EOC signal LOW (0)
Step 5 : Determine input mode
: Check sign bit
Step 4 : Read all (12) digital output (LSB and MSB)
Step 5 : If input mode negative
5.1 Perform two's compliment
Step 6: Convert binary value (digital output) to decimal value
Step 7: Store converted reading into shared memory
Step 8: End

Fig. 9. Analog Input Preprocessing algorithm

5.2 Data presentation

The *Interactive System* provides important information about the PESS system. The main goal of the *Interactive System* is to display current sensors' readings upon requested by the user. It also provides other information of the system (PESS) such as disk usage and memory status which viewed at the LCD panel. Another feature included in *Interactive System* is a control process. This process is to enable user to restart or shutdown the PESS for maintenance purposes. The matrix keypad functions as an input device that handles menu selection in the *Interactive System*. Fig. 10 outlines the main flow chart of the *Interactive System*.

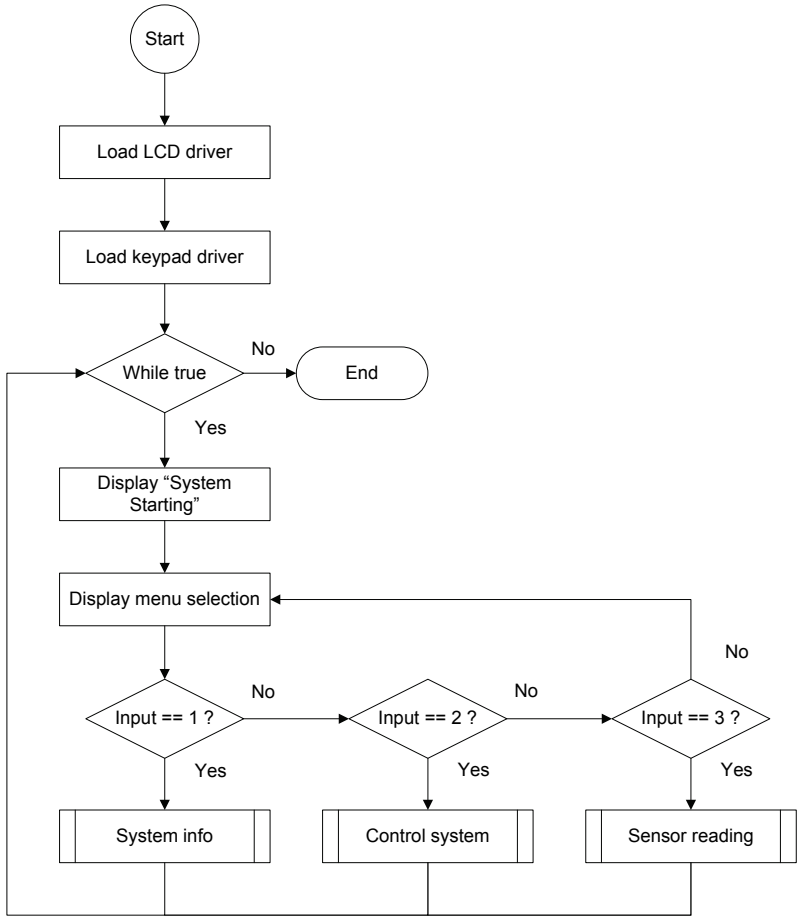


Fig. 10. *Interactive System* flowchart

Three options are provided: to check current sensors' readings, to check systems' information or to control the system. Three subroutines are created to handle those processes which are *system info*, *control system* and *sensor reading* as outlined by Fig. 10 above. Actually the processes of these three subroutines are carried out by combining the binary C code and shell scripts. Shell scripts retrieve current sensors' readings which are processed by the *sensor* program, and manipulate Linux commands to retrieve system information and control the system. The binary C code grabs the data given by the shell script codes and displays them.

6. PESS implementation

Standard method to gain the result of environment parameters such as water and air quality is using laboratory experiment. The laboratory experiment is not suitable for long period testing work such as in monitoring process. The alternatives method can resolve that

limitation. The US Environment Protection Agency (EPA) define alternatives method as any method but has been demonstrated in specific cases to produce results adequate for compliance monitoring (Quevauviller, 2006).
The alternatives method leads to real-time data sampling which can produce instant output result for *in situ* deployment. It also provides easier usage with advance electronic devices in a compact size but can perform multitasks excellently. The handheld instrument usage is one of the alternatives methods such as using Data Acquisition (DAQ) device. The DAQ device such as SBC offers variety of peripherals to make it function as a standalone system. Meanwhile the ion specific electrodes is also been used in many application with handheld instrument. For example, non-invasive chemical sensor arrays provide a suitable technique for *in situ* monitoring (Bourgeois, 2003). Many researches use specific ion selective electrode or sensor array for detection of target environmental substance or gases (Carotta, 2000; Becker, 2000; Wilson, 2001; Lee, 2001).
The measurement of the LM35DZ temperature sensor is done without connecting the signal conditioning circuit. The LM35DZ sensors are only given a power supply and grounding. The sensor' outputs are connected directly to ADC port of SBC during measurement. Fig. 11 shows the experimental setup to acquire ion selective electrode's reading. Three parts involve here are: (1) SBC, (2) Sensors (electrodes) and (3) Signal conditioning circuit. While the red arrows marks from point A and B are the input and output from signal conditioning circuit respectively. Sensor reading' results are presented in next section.

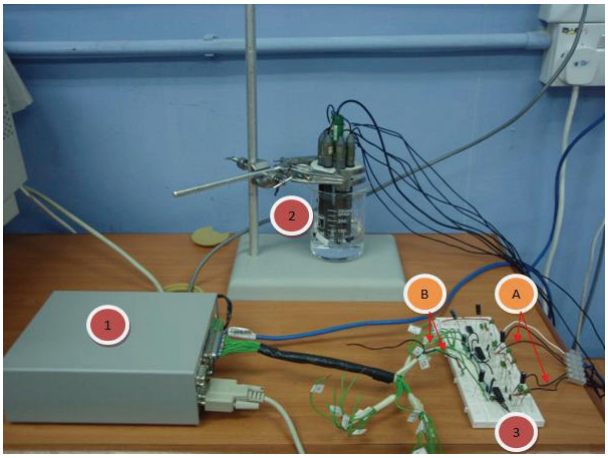


Fig. 11. Experimental setup of ion-selective electrodes

The programs called *sensor* and *Interactive System* are developed to handle all processes involved in Analog Input Preprocessing and Data Presentation modules respectively. Both modules are running separately but have a relationship in terms of data sharing. Fig. 12 outlines the state diagram for PESS system and the running processes listing. The current running process on PESS system including *sensor* and *Interactive System* as underlined in figure below. Analog Input Preprocessing module acquires data from sensors and storing converted data in a shared memory at PESS. These processes are repeated again with new inputs after certain time interval. While the *Interactive System* retrieve those converted data from shared memory and view it at LCD panel.

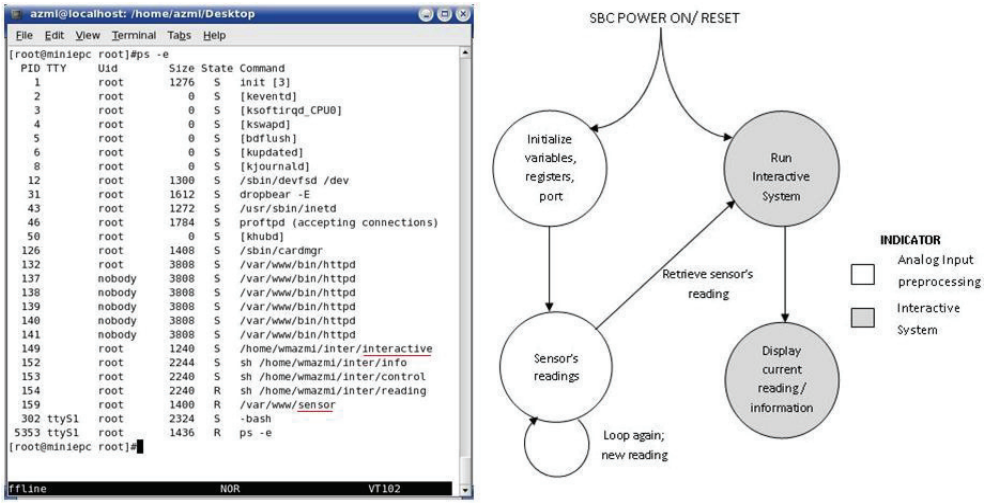

azmi@localhost: /home/azmi/Desktop
File Edit View Terminal Tabs Help
[root@miniepc root]#ps -e
PID TTY Uid Size State Command
1 root 1276 S init [3]
2 root 0 S [keventd]
3 root 0 S [ksoftirqd_CPU0]
4 root 0 S [kswapd]
5 root 0 S [bdflush]
6 root 0 S [kupdated]
8 root 0 S [kjournald]
12 root 1300 S /sbin/devfsd /dev
31 root 1612 S dropbear -E
43 root 1272 S /usr/sbin/inetd
46 root 1784 S proftpd (accepting connections)
50 root 0 S [khubd]
126 root 1408 S /sbin/cardmgr
132 root 3808 S /var/www/bin/httpd
137 nobody 3808 S /var/www/bin/httpd
138 nobody 3808 S /var/www/bin/httpd
139 nobody 3808 S /var/www/bin/httpd
140 nobody 3808 S /var/www/bin/httpd
141 nobody 3808 S /var/www/bin/httpd
149 root 1240 S /home/wmazmi/inter/interactive
152 root 2244 S sh /home/wmazmi/inter/info
153 root 2240 S sh /home/wmazmi/inter/control
154 root 2240 R sh /home/wmazmi/inter/reading
159 root 1400 R /var/www/sensor
302 ttyS1 root 2324 S -bash
5353 ttyS1 root 1436 R ps -e
[root@miniepc root]#
ffline NOR VT102
SBC POWER ON/ RESET
Initialize variables, registers, port
Run Interactive System
Retrieve sensor's reading
Sensor's readings
Display current reading/ information
Loop again; new reading
INDICATOR
Analog Input preprocessing
Interactive System

Fig. 12. PESS state diagram and running process listing

Four processes (programs) are set up to automatically start during the boot up program. The processes are: inserting the matrix keypad driver module; running *sensor* process; running the scripts (*info.sh, reading.sh* and *control.sh*) of *Interactive System*; and running the *Interactive System* program itself. These processes are underlined in Fig. 13. This procedure can be done by configuring how process will start up at */etc/init.d* directory.

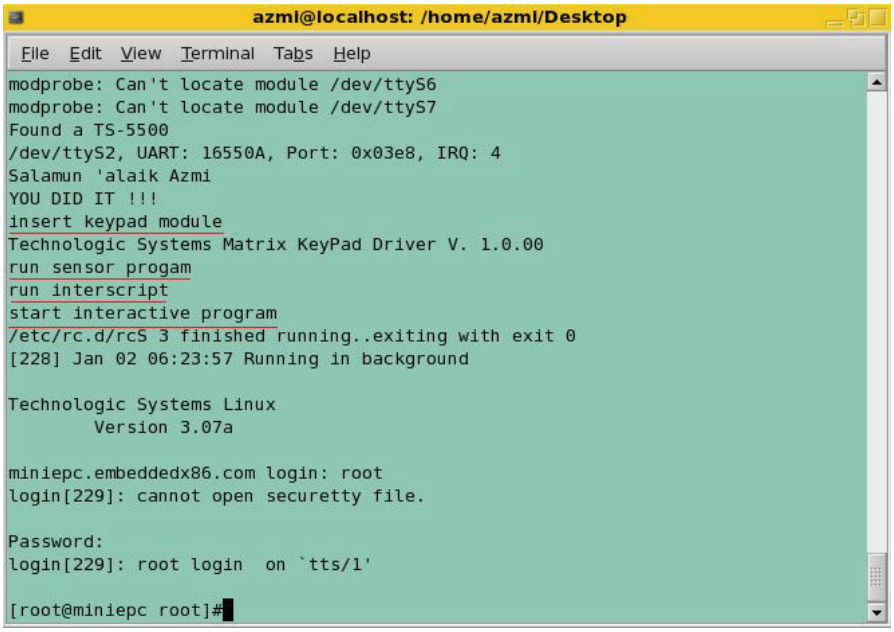

azmi@localhost: /home/azmi/Desktop
File Edit View Terminal Tabs Help
modprobe: Can't locate module /dev/ttyS6
modprobe: Can't locate module /dev/ttyS7
Found a TS-5500
/dev/ttyS2, UART: 16550A, Port: 0x03e8, IRQ: 4
Salamun 'alaik Azmi
YOU DID IT !!!
insert keypad module
Technologic Systems Matrix KeyPad Driver V. 1.0.00
run sensor progam
run interscript
start interactive program
/etc/rc.d/rcS 3 finished running..exiting with exit 0
[228] Jan 02 06:23:57 Running in background
Technologic Systems Linux
Version 3.07a
miniepc.embeddedx86.com login: root
login[229]: cannot open securetty file.
Password:
login[229]: root login on `tts/1'
[root@miniepc root]#

Fig. 13. Start processes automatically during system boot up